AUTOMATION OF THE PENSION MANAGEMENT SYSTEM

(A CASE STUDY OF THE PENSIONS DEPARTMENT, OFFICE OF THE HEAD OF THE CIVIL SERVICE OF THE FEDERATION)

BY

ZAINAB MAIKYAU
PGD/MCS/2005/1197

DEPARTMENT OF MATHEMATICS/COMPUTER SCIENCE
FEDERAL UNIVERSITY OF TECHNOLOGY MINNA

FEBRUARY 2008

AUTOMATION OF THE PENSION MANAGEMENT SYSTEM
(A Case Study of the Pensions Department, Office of the Head of the Civil Service of the Federation)

By

Zainab Maikyau
PGD/MCS/2005/1197

The Department Of Mathematics/Computer Science
Federal University of Technology Minna

A Project Submitted to
The School of Science and Science Education
In Partial Fulfillment of the Requirements for the Award of a
Postgraduate Diploma in Computer Science

February 2008

CERTIFICATION

This project titled, "Automation of the Pension Management System (A Case Study of the Office of the Head of the Civil Service of the Federation)", by Zainab Maikyau, meets the regulations governing the award of Postgraduate Diploma in Computer Science of Federal University of Technology, Minna.

....

Mallam Jiya Mohammed   Date

(Project Supervisor)

....

Dr N. I. Akinwande   Date

(Head of Department)

DEDICATION

This project is dedicated to my dear family,

for their unfailing love and unflinching

support.

ACKNOWLEDGEMENT

I wish to start by acknowledging our Father God Almighty for making this study a reality. Faithful Father, you are worthy!

My appreciation goes to my Project Supervisor Mr. M. Jiya for his useful and helpful counsel. I also wish to appreciate the Head of Department, Mathematics/Computer Science, Dr. N. I. Akinwande, the Post Graduate Diploma Coordinator, Mr. Ndanusa Abdulrahman, and our dedicated lecturers Mr. Abraham, Mr. Gbolahan, Mr. Onotu, Mr. Adamu and Mr. Shehu, just to mention but a few.

I wish to appreciate my office, the Office of the Head of the Civil Service of the Federation for granting me the permit to undertake this study, my Boss, Rev. Dr. E. A. Nwaise, and my colleagues in the office for their understanding and support. And also Mr. P. G. Atom for assisting me with the necessary materials I needed in The Office of the Head of the Civil Service of the Federation.

I wish to appreciate my dear husband, Barr. Y. C. Maikyau for his sponsorship and love, my lovely parents, Mr. and Mrs. U. Ahmed for their mentorship and encouragement, my dear sister, Habiba Sha'ibu for her support and care, my dear children Israel and Sarah for their prayers and patience, not forgetting my dear son King David who was born in the course of the study, you are my glory and my joy.

I wish to thank my kind hosts Aunty Abigail Jirgi, and Mr. and Mrs. Salman, for their help and hospitality. My dear friends, Funke Baruwa and Hajara Onoja, my church, Good Tidings Bible Church International, my fellow Children's Church teachers, my house help Esther, my nanny Lynda and my driver Marcus; I wish to say a big thank you.

God bless you all.

ABSTRACT

As a result of the increase in population ageing and a rise in the number of pensioners, collating, storing, modifying and updating pensioner's records and accounts has become a difficult task. Keeping of historical data and its development into a secure database is inevitable as the manual operation of any system has become a time consuming and labor intensive endeavor. The objectives of the pension department involve collating and updating pensioners' records and accounts, then using them to prepare pension payments. Prompt payment of pensions is of prime importance. The manual system is slow, cumbersome, time consuming, prone to errors and requires a good number of personnel due to large volume of data involved. Manual operation of the pension department delays preparation of pensioners' payments. This research work seeks to automate the manual pension management system of the pensions department so as to simplify daily routine. The feature will enable the user to create and store information into a database, modify and update records in the database and access the database to find specific information.

TABLE OF CONTENTS

CHAPTER ONE

GENERAL INTRODUCTION

1.1 Introduction

A pension management system is the totality of plans, procedures and legal processes of securing and setting aside funds to meet the social obligation of care which employers owe their employees on retirement or in case of death, and disability.

The objective of setting up a pension management system is therefore to provide employees of a corporate organization with a standard of living, which is reasonably consistent with that which, they enjoyed while in service.

A pension can be defined as a steady income given to a person (usually after retirement). Pensions are typically payments made in the form of a guaranteed annuity to a retired or disabled employee.

Pension plans can be divided into two broad types: defined benefit and defined contribution plans. Some plan designs combine characteristics of defined benefit and defined contribution types, and are often known as "hybrid" plans.

A defined benefit plan is a traditional pension plan that defines a benefit for an employee upon that employee's retirement. The benefit in this plan is determined by a formula that can incorporate the employee's pay, years of employment, age at retirement, and other factors.

The defined contribution plan provides for an individual account for each participant, and for benefits based solely on the amount contributed to the account, plus or minus income, gains, expenses and losses allocated to the account.

Hybrid plan designs combine the features of defined benefit and defined contribution plan designs. In general, they are usually treated as defined benefit plans for tax, accounting and regulatory purposes.

1.2 Problem Identification

A growing challenge for many nations is population ageing. As birth rates drop and life expectancy increases an ever-larger portion of the population is elderly. This leaves fewer workers for each retired person.

In almost all developed countries this means that government and public sector pensions could collapse their economies unless pension systems are reformed or taxes are increased. One method of reforming the pension system is to increase the retirement age.

As a result of this increase in population ageing and a rise in the number of pensioners, collating, storing, modifying and updating pensioner's records and accounts has become a difficult task.

Keeping of historical data and its development into a secure database is inevitable as the manual operation of any system has become a time consuming and labor intensive endeavour.

Currently, the trend to develop on line electronic calculators that replace traditionally complex spreadsheet calculations performed by Actuaries and Analysts is the industry norm in records management.

As such, the automation of the pension management system in the Pension Accounts Department of the Office of the Head of the Civil Service of the Federation has become absolutely necessary in recent times.

1.3 Aims and Objectives

This project seeks to create software that can:

i. Store pensioner's information in a secure database.

ii. Modify pensioner's records and accounts.

iii. Reduce the period of time used in compiling and preparing pensioner's pay by organizing frequent tasks and performing them without the need of a manual repetition.

iv. Reduce the man-hours needed for menial business because tasks can now be done automatically and are thereby less labour intensive.

v. Minimize the possible errors and mistakes in daily processes.

vi. Maximize the throughput of the organization by eliminating bottlenecks in daily processes.

vii. Simplify daily routine by optimizing complex and difficult tasks for maximum performance.

viii. Enhance cost effectiveness.

1.4 Scope and Limitations

Our scope of study is limited to the defined benefit plan pension management system used in the Pensions Department, of the Office of the Head of the Civil Service of the Federation which is our case study.

As stated earlier, the benefit of this plan is determined by a formula that incorporates the employee's pay, years of employment, age at retirement, grade level and rank at retirement and other factors.

The cost of this plan is very low for a young workforce, but extremely high for an older workforce. This age bias, the difficulty of portability and open ended risk, makes defined benefit plans better suited to large employers with less mobile workforces, such as the public sector.

The Federal Government of Nigeria has used this plan for several years until now. Under this plan, the government provides all the funds to operate the pension management system. As a result, public servants under this plan do not contribute towards their pensions at all. The government meets all retirement benefit liabilities of its retirees under this plan. The Pensions Department still uses this plan.

Recently however, the Government introduced the defined contribution plan and has set up The National Pension Commission to see to the execution of this plan. Thus, pensions are now based solely on the amount contributed to the account unlike the defined benefit plan pension management system used in the Pensions Department.

1.5 Significance of Study

This research work seeks to automate the manual pension management system of the Pensions Department so as to simplify daily routine. The feature will enable the user to create and store information into a database, modify and update records and accounts in the database and look through the database to find specific information.

An index will be used to keep the information in the database in alphabetical or numerical order automatically. Data can be printed for use by management or the end user in a useful manner. Storage, retrieval and organization will be done in an efficient manner.

CHAPTER TWO

LITERATURE REVIEW

2.1 Definition of Pension

A pension is a steady income given to a person (usually after retirement). Pensions are typically payments made in the form of a guaranteed annuity to a retired or disabled employee. Some retirement plan (or superannuation) designs accumulate a cash balance (through a variety of mechanisms) that a retiree can draw upon at retirement, rather than promising annuity payments. These are often also called pensions.

In either case, a pension created by an employer for the benefit of an employee is commonly referred to as an occupational or employer pension. Labor unions, the government, or other organizations may also fund pensions.

Occupational pensions are a form of deferred compensation, usually advantageous to employee and employer for tax reasons. Many pensions also contain an insurance aspect, since they often will pay benefits to survivors or disabled beneficiaries, while annuity income insures against the risk of longevity.

While other vehicles (certain lottery payouts, for example, or an annuity) may provide a similar stream of payments, the common use of the term pension is to describe the payments a person receives upon retirement, usually under pre-determined legal and/or contractual terms.

A pension scheme is the totality of plans, procedures and legal processes of securing and setting aside funds to meet the social obligation of care which employers owe their employees on retirement or in case of death, and disability.

The objective of setting up a pension scheme is therefore to provide employees of a corporate organization with a standard of living, which is reasonably consistent with that which, they enjoyed while in service.

It is interesting to note that the establishment of pension schemes is not new. In the African context pension was provided in a reciprocal situation whereby the young people are collectively taken care of and provided the necessary education to enable them take care of the elderly who at old age would be unable to provide for themselves. Thus, some form of Pension Schemes had been established for as long as there had been human race, although its development in modern times appears to be slow.

The Federal Government's interest and the first initiative on pension matters was the enactment of the Pension Ordinance of 1951 which had a retrospective effect from 1946. The Pensions Decree No. 102 of 1979, which is the extant law on pension matters, gave the Federal Government exclusive responsibility for pension matters in the whole Federation.

This explains why the Federal Government has taken necessary measures to establish and nurture pension schemes in the public service. Government provides all the funds to operate the schemes at the moment. As a result, public servants do not contribute towards their pensions at all prior to the introduction of the defined contribution plan.

The 1979 constitution of the Federal Republic of Nigeria made pension a first charge to the Consolidated Revenue Fund of the Federation, but the 1999 Federal Constitution has reversed this provision, making pension funding an annual budgetary affair.

This has greatly affected prompt pension payment as the total fund released may not be available as and when due. Under the defined benefit pay-as-you-go system which is being practiced, government will always have to meet retirement benefit liabilities of its retirees.

2.2 Types of Pensions

Retirement pension or superannuation plans are an arrangement in which an employer (for example, a corporation, labor union, and government agency) provides income to its employees after retirement. Pension plans are a form of "deferred compensation".

Pension plans can be divided into two broad types: defined benefit and defined contribution plans. Some plan designs combine characteristics of defined benefit and defined contribution types, and are often known as "hybrid" plans.

Defined Benefit Plans

A traditional pension plan that defines a benefit for an employee upon that employee's retirement is a defined benefit plan.The benefit in a defined benefit pension plan is determined by a formula that can incorporate the employee's pay, years of employment, age at retirement, and other factors.

Defined benefit pensions tend to be less portable than defined contribution plans even if the plan allows a lump sum cash benefit at termination due to the difficulty of valuing the transfer value.

On the other hand, defined benefit plans typically pay their benefits as an annuity, so retirees do not bear the investment risk of low returns on contributions or of outliving their retirement income. The open ended nature of this risk to the employer is the reason given by many employers for switching from defined benefit to defined contribution plans.

The cost of a defined benefit plan is very low for a young workforce, but extremely high for an older workforce. This age bias, the difficulty of portability and open ended risk, makes defined benefit plans better suited to large employers with less mobile workforces, such as the public sector.

The "cost" of a defined benefit plan is not easily calculated, and requires an actuary or actuarial software. However, even with the best of tools, the cost of a defined benefit plan will always be an estimate based on economic and financial assumptions.

These assumptions include the average retirement age and life span of the employees, the returns earned by the pension plan's investments and any additional taxes or levies. So, for this arrangement, the benefit is known but the contribution is unknown even when calculated by a professional.

Defined Contribution Plans

A defined contribution plan is a plan providing for an individual account for each participant, and for benefits based solely on the amount contributed to the account, plus or minus income, gains, expenses and losses allocated to the account.

Plan contributions are paid into an individual account for each member. The contributions are invested, for example in the stock market, and the returns on the investment (which may be positive or negative) are credited to the individual's account.

On retirement, the member's account is used to provide retirement benefits, often through the purchase of an annuity which provides a regular income. Defined contribution plans have become more widespread all over the world in recent years, and are now the dominant form of plan in the private sector in many countries.

For example, the number of defined benefit plans has been steadily declining, as more and more employers see the large pension contributions as a large expense that they can avoid by disbanding the plan and instead offering a defined contribution plan.Money contributed can either be from employee salary deferral or from employer contributions or matching. The portability of defined contribution pensions is legally no different from the portability of defined benefit plans.

However, because of the cost of administration and ease of determining the plan sponsor's liability for defined contribution plans (you don't need to pay an actuary tocalculate the lump sum equivalent that you do for defined benefit plans) in practice, defined contribution plans have become generally portable.

In a defined contribution plan, investment risk and investment rewards are assumed by each individual/employee/retiree and not by the sponsor/employer. In addition, participants do not typically purchase annuities with their savings upon retirement, and bear the risk of outliving their assets.

The "cost" of a defined contribution plan is readily calculated, but the benefit from a defined contribution plan depends upon the account balance at the time an employee is looking to use the assets. So, for this arrangement, the contribution is known but the benefit is unknown (until calculated).

Despite the fact that the participant in a defined contribution plan typically has control over investment decisions, the plan sponsor retains a significant degree of fiduciary responsibility over investment of plan assets, including the selection of investment options and administrative providers.

Hybrid and Cash Balance Plans

Hybrid plan designs combine the features of defined benefit and defined contribution plan designs. In general, they are usually treated as defined benefit plans for tax, accounting and regulatory purposes.

As with defined benefit plans, investment risk in hybrid designs is largely borne by the plan sponsor. As with defined contribution designs, planned benefits are expressed in the terms of a notional account balance. They are usually paid as cash balances upon termination of employment.

These features make them more portable than traditional defined benefit plans and perhaps more attractive to a more highly mobile workforce. A typical hybrid design is the Cash Balance Plan, where the employee's notional account balance grows by some defined rate of interest and annual employer contribution.

2.3 Financing Pensions

There are various ways in which a pension may be financed. In an unfunded defined benefit pension, no assets are set aside and the benefits are paid for by the employer or other pension sponsor as and when they are paid. Pension arrangements provided by the state in most countries in the world are unfunded, with benefits paid directly from current workers' contributions and taxes. This method of financing is known as Pay-as-you-go.

In a funded defined benefit arrangement, an actuary calculates the contributions that the plan sponsor must make to ensure that the pension fund will meet future payment obligations. This means that in a defined benefit pension, investment risk and investment rewards are typically assumed by the sponsor/employer and not by the individual. If a plan is not well-funded, the plan sponsor may not have the financial resources to continue funding the plan.

Defined contribution pensions, by definition, are funded, as the "guarantee" made to employees is that specified (defined) contributions will be made during an individual's working life. Pension reforms have gained pace worldwide in recent years and funded arrangements are likely to play an increasingly important role in delivering retirement income security and also affect securities markets in future years.

In the Office of the Head of the Civil Service of the Federation, preparing and submission of annual budget proposals for pension funds is the conventional procedure for obtaining financial assistance for an approved scheme. What is involved is to prepare a statement of demand based on the number of beneficaries and their entitlements on annual basis. the actuarial valuation of the scheme should be used in the preparation of the budget proposal.

The budget formats should contain the following information:

- Full Name of Retiree
- Date of Birth
- Date of First Appointment
- Date of Retirement
- Position Held on Retirement

- Total Length of Service
- Date of Commencement of Pension
- Pension Identity Card Number
- Salary Grade Level at Retirement
- Total Annual Pension Based on Salary
- Total Annual Pension Based on Total Emolument

The detailed information demanded is to satisfy the funding required to pay pensioners. The security of the funds would be brighter if Government is satisfied with the information provided. This procedure for sourcing funds from Government was approved to replace the practice where Insurance Companies and Insurance Brokers send their Demand Notes for funds directly to the Ministry of Finance.

It is expected that once this Office receives such budget proposals, they would be collated and forwarded to the Budget Office, Federal Ministry of Finance, which makes arbitrary allocation to government parastatals; and this has created the unstable pension situation of most parastatals, with huge unpaid liabilities.

The public service pension scheme is based on a defined benefit structure. If an officer retires from service and provided that he has served the qualifying period for penison, he shall be entitled to a lump sum gratuity from the pension fund and pension for life payable on monthly installments. At no time shall a public officer be required to make any form of contribution for his pension under this defined benefit structure.

2.4 Guidelines on Pension Schemes

Guidelines on the management and administration of pension schemes in the public service reflect the nature and type of schemes structure that is adopted by the organization. The guidelines for managing the scheme in the core Civil Service are different from those put in place for managing the schemes in the parastatals and other Government agencies.

(i) The Civil Service Pension Scheme:

The core Civil Service has the self administered type of pension scheme. Its characteristics are that it is non-contributory and it is not funded. The arm of Government that is statutorily responsible for pension matters, is, the Office of Establishment and Pensions, Office of the Head of the Civil Service of the Federation. The Pension Schemes of parastatals are managed by their various Pension Boards of Trustees (BOT). The administration and management of the schemes are guided by the Pension laws and other Government policies, which are announced from time to time in Establishments' Circulars.

(ii) Funds Management:

The Establishments and Pensions Office prepares annual pension budget on the basis of the number of retirees on which the release of funds is based. These funds are, in turn, released to other agencies in accordance with the provisions of the Financial Instructions and other specific rules on the disbursement of pension funds.

Funds are released to Federal Ministries and Extra Ministerial Departments to pay gratuities in line with the policy of decentralization of the payment of gratuity. All the laws and rules on the pension rights of Civil Servant and other

Public Officers are normally observed as provided in pension law, and terms of contract.

Some of the funds were used to pay directly the pension of Federal Pensioners who elected to collect their benefits in Lagos in the past. At the moment payment is centralized in the Office of the Head of the Civil Service of the Federation, Abuja. Reimbursement of federal share of pension and gratuity to the States and other Agencies, had consumed about 80% of the annual budget proposals in the past. This was to offset payments made on behalf of the Federal Government.

However, with the direct takeover of payment of pension to Federal Pensioners and State Pensioners with federal share, the amount disbursed to states as reimbursements would have substantially reduced. With time, it is expected that disbursement to states would be less than 10% of the annual budget allocation except for states that would be prepared to pay gratuity on behalf of Federal Government and later seek reimbursement.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis and Design

System analysis is a detailed study of the current system, leading to specifications of a new system. Analysis is a detailed study of various operations performed by a system and their relationships within and outside the system. It is the art of examining an existing system in order to modify the existing system or create a new system.

3.1.1 Analysis of the Existing System

While doing system analysis and design, it is safe, and even preferable, to assume we know nothing about the problem at hand. Thus in the first phase, it is our job mainly to ask questions, do research, and try to come up with a preliminary plan. We need to determine the organizations objectives. Then we can see how the problem being studied fits in with them.

i. To define the objectives of the organization, we are going to do the following:

ii. Read internal documents about the organization.

iii. Read external documents about the organization.

iv. Interview important executives within the company.

v. Interview key users.

vi. Interview people inside the organization.

vii. Interview clients or customers affected by the system.

From these sources, we will find out what the organization is supposed to be doing and to some extent, how well it is doing it. We will also try to understand the "organizational culture" the set of shared attitudes, values, goals, and practices that characterize the organization.

With a full understanding of the goals of the organization, we can now take a closer look at the specifics. We can leave the system as it is, improve it, or develop a new system. After defining the objectives of the organization, which includes collating and updating pensioners records and accounts, then using them to prepare pension payments. We can deduce that prompt payment of pensions is of prime importance.

We can see that the existing system is manually operated and as a result lacks effeciency. This is because the manual system is slow, cumbersome, time consuming, prone to errors and requires a good number of personnel due to large volume of data involved. Manual operation of the pension department delays preparation of pensioners' payments.

The existing system collates pensioners' records and accounts in a physical file. These files are very numerous as a large number of pensioners exist in the Federation. In the event of updating pensioners' records or accounts, these files must be consulted. Retrieval of these files for modification or certification is an upheaval task. A lot of time is used in the compilation and preparation of pensioner's pay.

Frequent tasks need a lot of a manual repetition. There are a lot of man-hours needed for menial business which makes work very labour intensive. Due to a lot of manual repetition, there are possible errors and mistakes in daily processes. Achieving the objectives of the organization becomes difficult as a lot of bottlenecks in daily processes exist. Daily routine is complex and a lot of difficult tasks exist.

3.2 System Design

Based on the user requirements and the detailed analysis of a new system, the new system must be designed. This is the phase of system designing. It is a most crucial phase in the development of a system.

The new system is an automated system which uses a database management system to create, store, modify and delete pensioners' records and accounts. It is an information system which aims at providing detailed information on a timely basis throughout the organisation so as to help management take proper and effective decisions such as ensuring the prompt and efficient payment of pensions in this case.

The information system cuts across departmental lines and helps in achieving overall optimization for the organisation. It will be an integrated information system which is based on the presumption that data and information are used by more than one system in the organisation and accordingly, the data and information are channeled into a reservoir or database. All the data processing and provision of information is derived and taken from this common database.

The multiuser database access is used for the new system. The multiuser database is a shared or centralized database which uses the client/server network shared by many users in one company or organization in one location. The organization owns the database which is stored on a server. Users on computers with client software are linked to the database via a client/server arrangement on a local area network (LAN).

The local area network (LAN) is a communications network that connects computers and devices in a limited geographical area, such as one office, one building, or a group of buildings close together (for instance, a college campus). LANs have replaced large computers for many functions and are considerably less expensive.

The client/server network has three distinct components:

1. The database server which is the "back end" that manages the database, efficiently and optimally among various clients that simultaneously request the server for the same resource.

2. The client application which is the "front end" that users apply to so as to interact with data. It presents an interface between the user and the resource to complete the job.

3. The third component is the network. Both the client and the server run communication software that allows them to talk across the network. The communication software are the vehicles that transmit data between the clients and the server in the client/server system.

3.3 Programming Language

A programming language is a natural system for communicating with the computer. It is a means by which users express what they want the computer to do for them.

3.3.1 Choice of Programming Language

The C# Language

C# is a type-safe, component-based, high-performance language that is designed for the Microsoft .NET framework. C# 2.0 is the new generation of C#, upgraded with Generics and other advanced features and fully integrated into .NET 2.0 and Visual Studio 2005. If you are developing Windows or web applications or web services for the .NET platform, C# is in many ways the language of choice.

The goal of C# 2.0 is to provide a mature, simple, safe, modern, object-oriented, internet-centric, high-performance language for .NET development. In much the same way that you can see in young children the features and personalities of their parents and grandparents, you can easily see in C# the influence of Java, C++, Visual Basic (VB), and other languages, but you can also see the lessons learned since C# was first introduced.

The .NET Platform

The .NET platform is a development framework that provides a new application programming interface (API) to the services and APIs of classic Windows operating systems while bringing together a number of disparate technologies that emerged from Microsoft during the late 1990s. This includes COM+ component services, a commitment to XML and object-oriented design, support for standardized protocols such as SOAP, WSDL, and UDDI, and a focus on the internet.

The Common Language Runtime

The most important component of the .NET Framework is the Common Language Runtime (CLR), which provides the environment in which programs are executed. The CLR includes a virtual machine that creates, manages, secures and cleans up after objects. The CLR is a set of framework classes that include a layer for data (ADO.NET) and for XML. On top of all this, the CLR creates a layer for three types of applications: Windows desktop applications, web Applications and web services (see future article, "What Are Web Services?").

Compilation and the Microsoft Intermediate Language

As noted earlier, .NET Windows and web programs are not compiled into executable files, they are compiled into Assemblies that consist of Microsoft Intermediate Language (MSIL) instructions. The IL is saved in a file on disk. When you run your program, the IL is compiled again, using the Just In Time (JIT) compiler (a process often called JITing). The result is machine code, executed by the machine's processor.

3.4 Input Design

A database management system will be used to organize pensioner's data into easy to use information. A database management system (DBMS), or data manager, refers to software written specifically to control the structure of a database and access to the data.

The database should include the following information about the pensioner:

Last Name	Date of Retirement	Annual Pay	Bank
First Name	Grade Level	Gross Pay	Account Number
Birthday	Last Department	Monthly Pay	Control Number
Date of Employment	Rank	Arrears	Picture

Control bottons should include:

Previous	Search	Add New
Next	Delete	Save
Get All	Edit	Print

3.5 Output Design

i. To enter a new record into the database, the **Add New** button should be clicked, and then the pensioner's record entered with all necessary details for the database. After entering the new record, the **Save** button should be used to save the new record.

ii. To modify an existing record, the **Edit** button should be clicked and the surname of the pensioner entered. After locating the pensioner, all necessary modifications can be done and then saved, using the **Save** button.

iii. To locate a pensioner's record, the **Search** button can be used for individual records, while the **Previous** button can be used to get previous pensioner's record, and the **Next** button can be used to get the next pensioner's record in the database. The **Get All** button can be used to get all the records of pensioners available in the database.

iv. In the event of a dead pensioner or duplicated pensioner's records, the **Delete** button is used to delete such records.

v. To select a bank, use the drop down button and select one from the list of options.

vii. To select a date, a calendar is available when you click the drop down button on the date of birth, date of employment or date of retirement.

viii. When a hard copy of a pensioner's record is needed, it can easily be printed out by using the **Print** button.

CHAPTER FOUR

SOFTWARE DESIGN AND IMPLEMENTATION

4.1 Hardware Requirements

To meet our desired objective, that is, the Automation of the Pensions Department of the Office of the Head of the Civil Service of the Federation, the following hardware requirements are needed:

1. Pentium IV Processor
2. 512 Megabytes Of RAM
3. Computer Systems With CD-ROM And 3.5 Drives
4. Printers
5. Power Stabilizers
6. Uninterrupted Power Supply (UPS).

4.2 Software Requirements

The software requirements include:

1. Windows Operating System for remote and stand alone computers.
2. Windows Net work server for the host computer.
3. dotNetFramework & Windows Installer 3.1
4. Source Code

4.3 Changeover Procedure

File conversion is a vital activity which is sometimes under estimated. It is the process of converting the old file data into the form required by the new system. Changeover from the old to the new system may take place when the new system has been tested, and there is general satisfaction with the results of the test.

4.3.1 Choice of Changeover Procedure

Changeover may be achieved in a number of ways, as several changeover procedures exist. The most common methods are:

- Direct changeover
- Parallel changeover
- Pilot changeover
- Stage changeover

Direct changeover is the bold move which should be undertaken when everyone involved has confidence in the new system. When a direct changeover is planned, system test and training should be comprehensive and the changeover itself planned in detail. This method is potentially the least expensive but the most risky.

Parallel changeover means processing the old and new system side by side so as to cross-check the result. Its main attraction is that the old system is kept alive and in cooperation with the new system until the new system has been satisfactorily proven beyond all reasonable doubt.

Pilot changeover is similar to parallel changeover. Data from one or more previous period, for the whole or part of the system is run on the new system after results are compared with the old. This method is more like an extended system test but it may be considered a more practical form of changeover for organizational reasons.

Stage changeover involves a series of limited size direct changeovers. The new system is being introduced piece by piece. A complete part or logical section is committed to the new system while the remaining parts or sections are processed by the old system. Only when the selected part is operating satisfactorily is the remainder transferred.

The choice of changeover procedure to be used here is the parallel changeover procedure. This system allows the result of the new system to be compared with the old system before acceptance by the users thereby promoting user confidence.

The only demerit of this method is the attendance cost of running two systems at the same time. However, it is the most suitable for the Pensions Department as we cannot afford to delay the pensioners pay for even a single month.

4.4 Cost and Benefit Analysis

The changes or absence of changes will have a price tag. Is the existing technology cost effective? Will the cost be offset by the benefits? Cost may depend on benefits, which may offer savings.

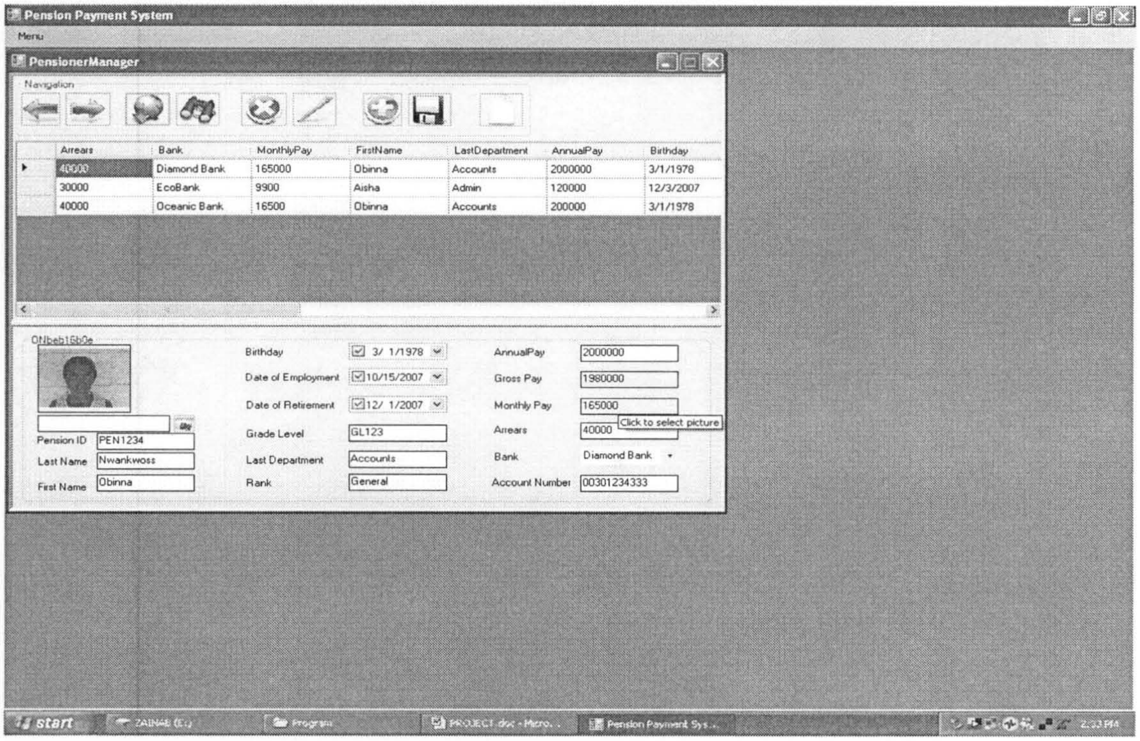
The old system is slow, cumbersome, time consuming, prone to errors and requires a good number of personnel due to large volume of data involved. Manual operation of the pension department delays preparation of pensioners' payments.

The benefits of the data base management system cannot be overemphasized. They include reduced data redundancy, improved data integrity, increased security and ease of data maintenance.

The table below shows the cost of implementing the new system.

Requirements	**Quantity Required**	**Cost (₦)**
Hardware		
1. Computer Systems with Pentium IV Processor, 512 Megabytes Of RAM, CD-ROM and 3.5 Drives	10	300,000.00
2. Printers	2	50,000.00
3. Power Stabilizers	2	20,000.00
4. Uninterrupted Power Supply (UPS)	1	50,000.00
Software		
1. Windows Operating System for remote and stand alone computers.	1	10,000.00
2. Windows Net work server for the host computer.	1	10,000.00
3. dotNetFramework and Windows Installer 3.1	10	5,000.00
4. Source Code	1	5,000.00
Others		
1. Training	Personnel	100,000.00
2. Maintenance	New System	100,000.00
3. Networking	Entire Workgroup	50,000.00
Total Cost		700,000.00

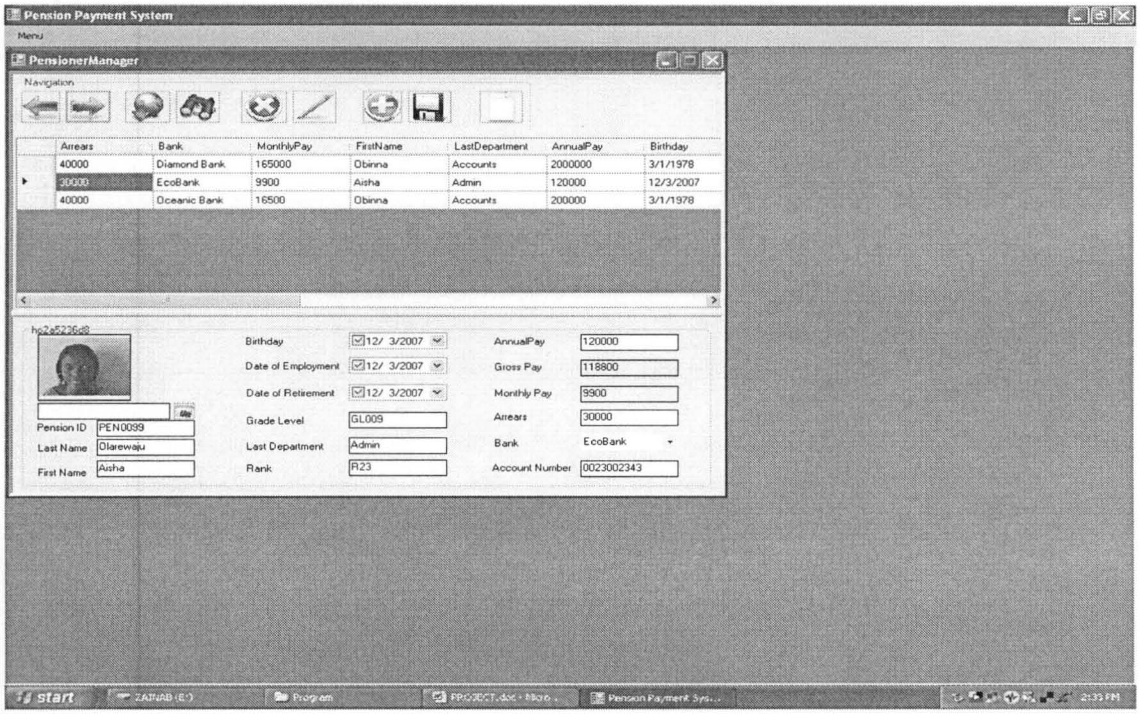
4.5 Program Output



	Arrears	Bank	MonthlyPay	FirstName	LastDepartment	AnnualPay	Birthday
▸	40000	Diamond Bank	165000	Obinna	Accounts	2000000	3/1/1978
	30000	EcoBank	9900	Aisha	Admin	120000	12/3/2007
	40000	Oceanic Bank	16500	Obinna	Accounts	200000	3/1/1978

0Nbeb16b0e

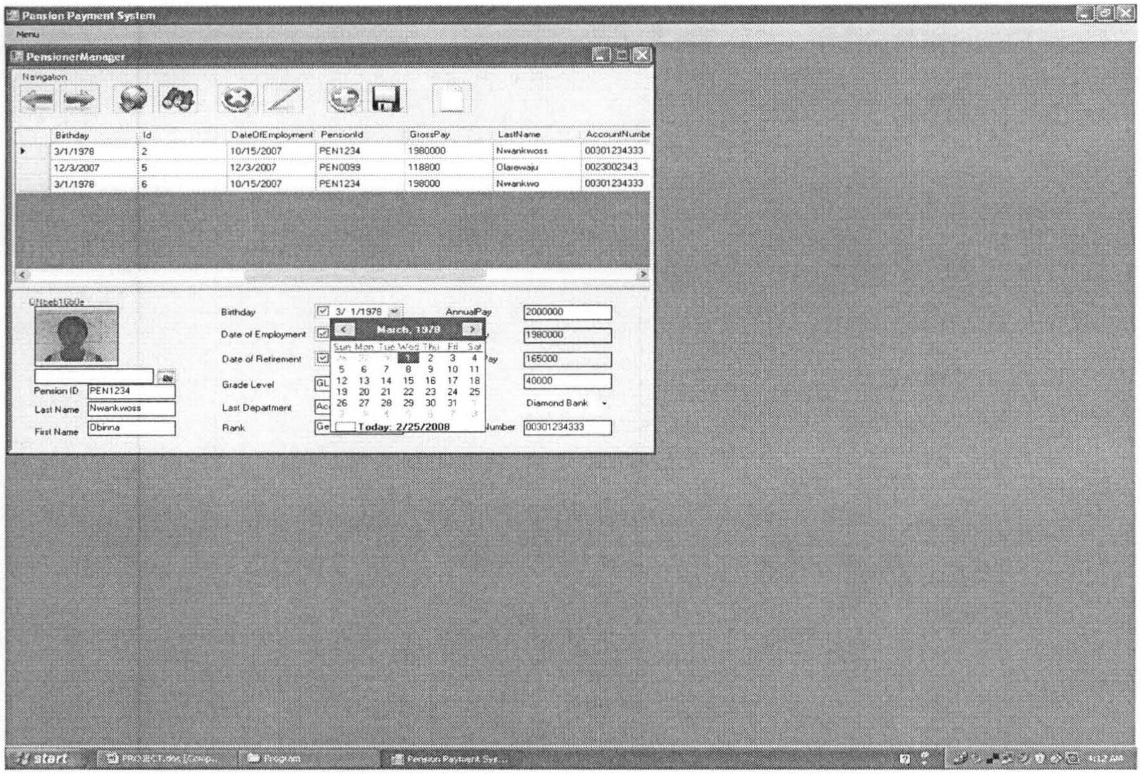
Pension ID: PEN1234
Last Name: Nwankwoss
First Name: Obinna
Birthday: ☑ 3/ 1/1978
Date of Employment: ☑ 10/15/2007
Date of Retirement: ☑ 12/ 1/2007
Grade Level: GL123
Last Department: Accounts
Rank: General
AnnualPay: 2000000
Gross Pay: 1980000
Monthly Pay: 165000
Arrears: 40000
Bank: Diamond Bank
Account Number: 00301234333



	Arrears	Bank	MonthlyPay	FirstName	LastDepartment	AnnualPay	Birthday
	40000	Diamond Bank	165000	Obinna	Accounts	2000000	3/1/1978
▸	30000	EcoBank	9900	Aisha	Admin	120000	12/3/2007
	40000	Oceanic Bank	16500	Obinna	Accounts	200000	3/1/1978

hc2a5236d8

Pension ID: PEN0099
Last Name: Olarewaju
First Name: Aisha
Birthday: ☑ 12/ 3/2007
Date of Employment: ☑ 12/ 3/2007
Date of Retirement: ☑ 12/ 3/2007
Grade Level: GL009
Last Department: Admin
Rank: R23
AnnualPay: 120000
Gross Pay: 118800
Monthly Pay: 9900
Arrears: 30000
Bank: EcoBank
Account Number: 0023002343



	Birthday	Id	DateOfEmployment	PensionId	GrossPay	LastName	AccountNumbe
▸	3/1/1978	2	10/15/2007	PEN1234	1980000	Nwankwoss	00301234333
	12/3/2007	5	12/3/2007	PEN0099	118800	Olarewaju	0023002343
	3/1/1978	6	10/15/2007	PEN1234	198000	Nwankwo	00301234333

Pension ID PEN1234
Last Name Nwankwoss
First Name Obinna

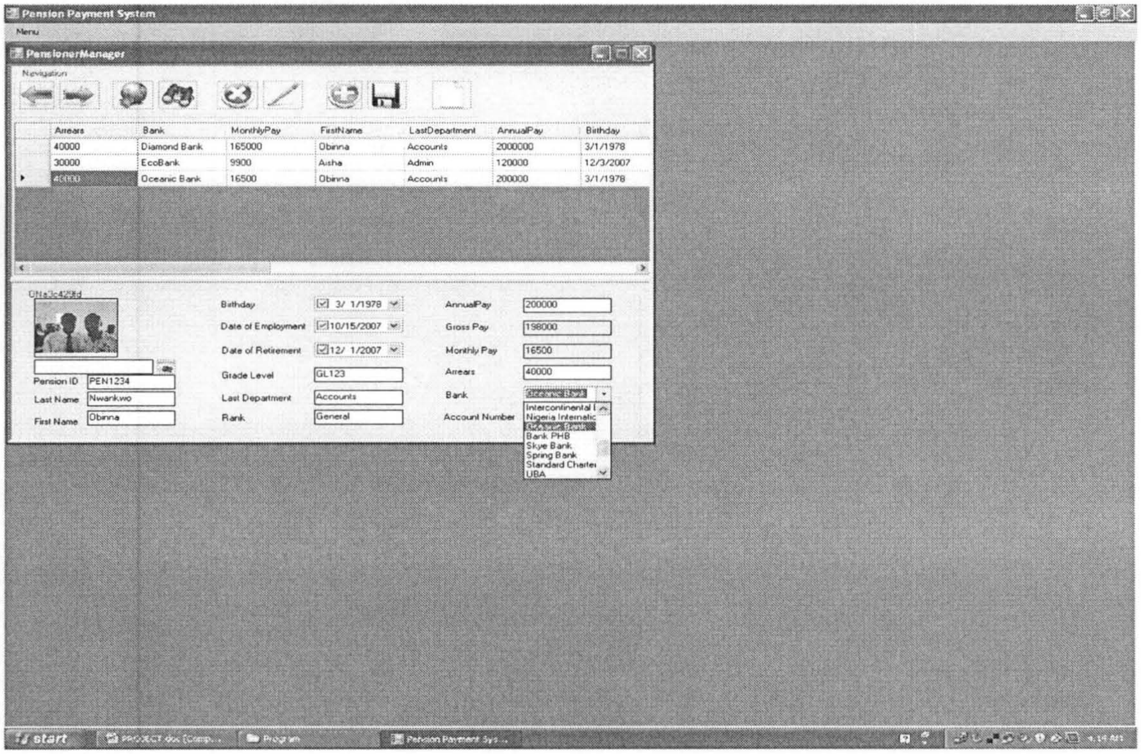
Birthday ☑ 3/ 1/1978
Date of Employment ☑
Date of Retirement ☑
Grade Level GL
Last Department Ac
Rank Ge

March, 1978

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Today: 2/25/2008

AnnualPay 2000000
1980000
ay 165000
40000
Diamond Bank
umber 00301234333



	Arrears	Bank	MonthlyPay	FirstName	LastDepartment	AnnualPay	Birthday
	40000	Diamond Bank	165000	Obinna	Accounts	2000000	3/1/1978
	30000	EcoBank	9900	Aisha	Admin	120000	12/3/2007
▸	40000	Oceanic Bank	16500	Obinna	Accounts	200000	3/1/1978

Pension ID PEN1234
Last Name Nwankwo
First Name Obinna

Birthday ☑ 3/ 1/1978
Date of Employment ☑10/15/2007
Date of Retirement ☑12/ 1/2007
Grade Level GL123
Last Department Accounts
Rank General

AnnualPay 200000
Gross Pay 198000
Monthly Pay 16500
Arrears 40000
Bank Oceanic Bank
Account Number

Interconnental I
Nigeria Internatic
Oceanic Bank
Bank PHB
Skye Bank
Spring Bank
Standard Charte
UBA

CHAPTER FIVE

SUMMARY/RECOMMENDATION

5.1 Summaries

The objective of setting up a pension management system is to provide employees of a corporate organization with a standard of living, which is reasonably consistent with that which, they enjoyed while in service.

As a result of the increase in population ageing and a rise in the number of pensioners, collating, storing, modifying and updating pensioner's records and accounts has become a difficult task. Keeping of historical data and its development into a secure database is inevitable as the manual operation of any system has become a time consuming and labor intensive endeavor.

The objectives of the pensions department involve collating and updating pensioners' records and accounts, then using them to prepare pension payments. Prompt payment of pensions is of prime importance. The manual system is slow, cumbersome, time consuming, prone to errors and requires a good number of personnel due to large volume of data involved. Manual operation of the pension department delays preparation of pensioners' payments.

This research work seeks to automate the manual pension management system of the pensions department so as to simplify daily routine. The feature will enable the user to create and store information into a database, modify and update records in the database and access the database to find specific information.

A database management system is being used to organize pensioner's data into easy to use information. A database management system, or data manager, refers to software written specifically to control the structure of a database and access to the data.

The new system is an integrated information system which is based on the presumption that data and information are used by more than one system in the organisation and accordingly, the data and information are channeled into a reservoir or database.

The multiuser database which is a shared or centralized database that uses the client/server network shared by many users in one company or organization in one location is being used.

The C# programming language is used. C# is a type-safe, component-based, high-performance language that is designed for the Microsoft .NET framework. C# 2.0 is the new generation of C#, upgraded with Generics and other advanced features and fully integrated into .NET 2.0 and Visual Studio 2005.

5.2 Conclusions

After a successful parallel run of the old and the new system, it has been clearly seen that the new automated pension management system which uses the database management system is obviously faster at collating pensioners' records and accounts. It is also easier to use and less labour intensive. Prompt payment of pensions is ensured as pensioners' records and accounts are better organised and easier to access.

The benefits of the data base management system cannot be overemphasized. They include the following: reduced data redundancy, improved data integrity, increased security and ease of data maintenance.

Data redundancy, or repetition, means that the same data fields (a person's address, say) appear over and over again in different files and often in different formats. In the old file processing system, separate files would repeat the same data, wasting storage space.

In a database mangement system, the information appears just once, freeing up more storage space. In the old data storage systems, if one field needed to be updated, someone had to make sure that it was updated in all the places it appeared - an invitation to error.

Data integrity means that data is accurate, consistent, and up to date. In the old system, when a change was made in one file, it might not have been made in other necessary files. The result was that some reports were produced with erroneous information.

In a DBMS, reduced redundancy increases the chances of data integrity - the chances that the data is accurate, consistent, and up to date - because each updating change is made in only one place.

Although various departments may share data, access to specific information can be limited to selected users. Thus, through the use of passwords, a pensioner's information in a pension management system database is made available only to those who have a legitimate need to know.

Database mangement systems offer standard procedures for adding, editing, and deleting records, as well as validation checks to ensure that the appropriate type of data is being entered properly and completely into each field type. Data backup utilities ensure availability of data in case of primary system failure.

The process is speeded up, and streamlined through elimination of unnecessary steps. Input errors or redundant output are reduced. Systems and subsystems are better integrated. Users are happier with the system and clients interact better with the system.

5.3 Recommendations

The development of the new system requires a long-term overall plan, commitment from management at all levels, highly technical personnel, availability of sufficient fund, and sophisticated technology. It also requires adequate standby facilities, without which the system is doomed to failure.

There is a need to acquire hardware and software and test the systems. It is also necessary to convert the hardware, software and files to the new system and train the users. There is also a need to audit the system, request feedback from its users, and evaluate it periodically.

Training is necessary so that people can understand their responsibility in making the system work. Two groups of people will need to be trained. The technical personnel who will develop, operate and maintain the system, and the systems user and their managers.

Training is done with a variety of tools. They run from documentation (instruction manuals) to videotapes, to live classes, to one-on-one/side-by-side teacher-student training. It is sometimes done by the organization's own staffers, and at other times, it is contracted out.

Even with the conversion accomplished and the users trained, the system won't run itself. There is need for system maintenance. System maintenance adjusts and improves the system by having system audits, user feedback, and periodic evaluations and by making changes based on new conditons.

References:

1. LUCAS, H.C. (1985) The Analysis, Design and
Implementation of Systems
McGraw Hill Publication, Singapore.

2. Pension Act 102 of 1979

3. Pension Reform Act 2004

4. Pension Trust Deeds for Parastatals

5. SENN, J. A. (1989) Analysis and Design of Information Systems
McGraw Hill Publications, Singapore.

6. Wikipedia, the Free Encyclopedia: Pension

7. WILLIAM, B. K. (1938) Using Information Technology:
A Practical Introduction to Computers and
Communications Complete Version
Brian K. Williams and Stacey C. Sawyer (2005)
6th Edition
McGraw Hill Technology Education

8. Working Circulars/Regulations on Pensions
Issued by Pension and Records Department
Establishment and Pensions Office
Office of the Head of the Civil Service of the Federation
Abuja, Nigeria.

Appendix

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Data.OleDb;

namespace PensionPaymentSystem
{
   public class Pensioner : DALBase
   {
      #region Properties and Fields
      //
      private int id;

      public int Id
      {
         get { return id; }
         set { id = value; }
      }
      private string pensionId;

      public string PensionId
      {
         get { return pensionId; }
         set { pensionId = value; }
      }
      private string lastName;

      public string LastName
      {
         get { return lastName; }
         set { lastName = value; }
      }
      private string firstName;

      public string FirstName
      {
         get { return firstName; }
         set { firstName = value; }
      }
      private string birthday;

      public string Birthday
      {
         get { return birthday; }
         set { birthday = value; }
      }
      private string dateOfEmployment;

      public string DateOfEmployment
      {
         get { return dateOfEmployment; }
         set { dateOfEmployment = value; }
      }
      private string dateOfRetirement;

      public string DateOfRetirement
      {
         get { return dateOfRetirement; }
         set { dateOfRetirement = value; }
}
private string gradeLevel;

public string GradeLevel
{
    get { return gradeLevel; }
    set { gradeLevel = value; }
}
private string lastDepartment;

public string LastDepartment
{
    get { return lastDepartment; }
    set { lastDepartment = value; }
}
private string rank;

public string Rank
{
    get { return rank; }
    set { rank = value; }
}
private string annualPay = "0";

public string AnnualPay
{
    get { return annualPay; }
    set { annualPay = value; }
}

public string GrossPay
{
    get
    {
        return CalculateGrossPay(double.Parse(this.annualPay)).ToString();
    }
}

public string MonthlyPay
{
    get
    {
        return CalculateMonthlyPay(double.Parse(this.GrossPay)).ToString();
    }
}
private string arrears;

public string Arrears
{
    get { return arrears; }
    set { arrears = value; }
}
private string bank;

public string Bank
{
    get { return bank; }
    set { bank = value; }
}
private string accountNumber;

public string AccountNumber
```

```
{
    get { return accountNumber; }
    set { accountNumber = value; }
}
private string controlNumber;

public string ControlNumber
{
    get { return controlNumber; }
    set { controlNumber = value; }
}

private string picture;

public string Picture
{
    get { return picture; }
    set { picture = value; }
}
//
#endregion

public static double CalculateGrossPay(double annualPy)
{
    return annualPy * .99;
}

public static double CalculateMonthlyPay(double grossPy)
{
    return grossPy / 12d;
}

public void Retrieve(string id, bool useControlId)
{
    OleDbDataReader reader = null;
    try
    {
        string column = useControlId ? "ControlNumber" : "Id";
        id = useControlId ? "'" + id + "'" : id;
        string cmdText = "select * from Pensioners where " + column + " = " + id;
        reader = base.GetReader(base.GetCommand(cmdText));
        while (reader.Read())
        {
            this.id = reader.GetInt32(0);
            this.pensionId = reader.GetString(1);
            this.lastName = reader.GetString(2);
            this.firstName = reader.GetString(3);
            this.birthday = reader.GetString(4);
            this.dateOfEmployment = reader.GetString(5);
            this.dateOfRetirement = reader.GetString(6);
            this.gradeLevel = reader.GetString(7);
            this.lastDepartment = reader.GetString(8);
            this.rank = reader.GetString(9);
            this.annualPay = reader.GetString(10);
            //this.grossPay = reader.GetString(11);
            //this.monthlyPay = reader.GetString(12);
            this.arrears = reader.GetString(13);
            this.bank = reader.GetString(14);
            this.accountNumber = reader.GetString(15);
            this.controlNumber = reader.GetString(16);
            this.picture = reader.GetValue(17) != DBNull.Value ? reader.GetString(17) : null;
        }
```

```
            }
            finally
            {
                if (reader != null)
                    reader.Close();
            }
        }

        public static List<Pensioner> Get(string whereClause)
        {
            OleDbDataReader reader = null;
            List<Pensioner> list = new List<Pensioner>();
            try
            {
                string cmdText = "select * from Pensioners ";
                if (whereClause.Length > 0)
                    cmdText += "where " + whereClause;
                reader = DALBase.GetReaderStatic(DALBase.GetCommandStatic(cmdText));
                if (reader.HasRows)
                {
                    while (reader.Read())
                    {
                        Pensioner pensioner = new Pensioner();
                        pensioner.Id = reader.GetInt32(0);
                        pensioner.pensionId = reader.GetString(1);
                        pensioner.lastName = reader.GetString(2);
                        pensioner.firstName = reader.GetString(3);
                        pensioner.birthday = reader.GetString(4);
                        pensioner.dateOfEmployment = reader.GetString(5);
                        pensioner.dateOfRetirement = reader.GetString(6);
                        pensioner.gradeLevel = reader.GetString(7);
                        pensioner.lastDepartment = reader.GetString(8);
                        pensioner.rank = reader.GetString(9);
                        pensioner.annualPay = reader.GetString(10);
                        //pensioner.grossPay = reader.GetString(11);
                        //pensioner.monthlyPay = reader.GetString(12);
                        pensioner.arrears = reader.GetString(13);
                        pensioner.bank = reader.GetString(14);
                        pensioner.accountNumber = reader.GetString(15);
                        pensioner.controlNumber = reader.GetString(16);
                        pensioner.Picture = reader.GetValue(17) != DBNull.Value ? reader.GetString(17) : null;
                        list.Add(pensioner);
                    }
                }
            }
            finally
            {
                if (reader != null)
                    reader.Close();
            }
            return list;
        }

        public void Save()
        {
            StringBuilder cmdText = new StringBuilder();
            if (this.id == 0)
            {
                //insert
                cmdText.Append("insert into Pensioners (PensionId, LastName, FirstName, Birthday, " +
                    "DateOfEmployment, DateOfRetirement, GradeLevel, LastDepartment, Rank, AnnualPay,
GrossPay," +
        " MonthlyPay, Arrears, Bank, AccountNumber, ControlNumber, PicturePath) values (");
    cmdText.Append(base.BuildStringParameter(this.pensionId));
    cmdText.Append(", ");
    cmdText.Append(base.BuildStringParameter(this.lastName));
    cmdText.Append(", ");
    cmdText.Append(base.BuildStringParameter(this.firstName));
    cmdText.Append(", ");
    cmdText.Append(base.BuildStringParameter(this.birthday));
    cmdText.Append(", ");
    cmdText.Append(base.BuildStringParameter(this.dateOfEmployment));
    cmdText.Append(", ");
    cmdText.Append(base.BuildStringParameter(this.dateOfRetirement));
    cmdText.Append(", ");
    cmdText.Append(base.BuildStringParameter(this.gradeLevel));
    cmdText.Append(", ");
    cmdText.Append(base.BuildStringParameter(this.lastDepartment));
    cmdText.Append(", ");
    cmdText.Append(base.BuildStringParameter(this.rank));
    cmdText.Append(", ");
    cmdText.Append(base.BuildStringParameter(this.annualPay));
    cmdText.Append(", ");
    cmdText.Append(base.BuildStringParameter(this.GrossPay));
    cmdText.Append(", ");
    cmdText.Append(base.BuildStringParameter(this.MonthlyPay));
    cmdText.Append(", ");
    cmdText.Append(base.BuildStringParameter(this.arrears));
    cmdText.Append(", ");
    cmdText.Append(base.BuildStringParameter(this.bank));
    cmdText.Append(", ");
    cmdText.Append(base.BuildStringParameter(this.accountNumber));
    cmdText.Append(", ");
    cmdText.Append(base.BuildStringParameter(this.controlNumber));
    cmdText.Append(", ");
    cmdText.Append(base.BuildStringParameter(this.picture));
    //cmdText.Append(", ");
    //cmdText.Append(base.BuildStringParameter(personnel.Nationality));
    cmdText.Append(")");
}
else
{
    //update
    cmdText.Append("update Pensioners set ");
    cmdText.Append("PensionId = ");
    cmdText.Append(base.BuildStringParameter(this.pensionId));
    cmdText.Append(", ");
    cmdText.Append("LastName = ");
    cmdText.Append(base.BuildStringParameter(this.lastName));
    cmdText.Append(", ");
    cmdText.Append("FirstName = ");
    cmdText.Append(base.BuildStringParameter(this.firstName));
    cmdText.Append(", ");
    cmdText.Append("Birthday = ");
    cmdText.Append(base.BuildStringParameter(this.birthday));
    cmdText.Append(", ");
    cmdText.Append("DateOfEmployment = ");
    cmdText.Append(base.BuildStringParameter(this.dateOfEmployment));
    cmdText.Append(", ");
    cmdText.Append("DateOfRetirement = ");
    cmdText.Append(base.BuildStringParameter(this.dateOfRetirement));
    cmdText.Append(", ");
    cmdText.Append("GradeLevel = ");
    cmdText.Append(base.BuildStringParameter(this.gradeLevel));
```

```
        cmdText.Append(", ");
        cmdText.Append("LastDepartment = ");
        cmdText.Append(base.BuildStringParameter(this.lastDepartment));
        cmdText.Append(", ");
        cmdText.Append("Rank = ");
        cmdText.Append(base.BuildStringParameter(this.rank));
        cmdText.Append(", ");
        cmdText.Append("AnnualPay = ");
        cmdText.Append(base.BuildStringParameter(this.annualPay));
        cmdText.Append(", ");
        cmdText.Append("GrossPay = ");
        cmdText.Append(base.BuildStringParameter(this.GrossPay));
        cmdText.Append(", ");
        cmdText.Append("MonthlyPay = ");
        cmdText.Append(base.BuildStringParameter(this.MonthlyPay));
        cmdText.Append(", ");
        cmdText.Append("Arrears = ");
        cmdText.Append(base.BuildStringParameter(this.arrears));
        cmdText.Append(", ");
        cmdText.Append("Bank = ");
        cmdText.Append(base.BuildStringParameter(this.bank));
        cmdText.Append(", ");
        cmdText.Append("AccountNumber = ");
        cmdText.Append(base.BuildStringParameter(this.accountNumber));
        cmdText.Append(", ");
        cmdText.Append("ControlNumber = ");
        cmdText.Append(base.BuildStringParameter(this.controlNumber));
        cmdText.Append(", ");
        cmdText.Append("PicturePath = ");
        cmdText.Append(base.BuildStringParameter(this.picture));
        //cmdText.Append(", ");
        //cmdText.Append("Id = ");
        //cmdText.Append(this.id);
        cmdText.Append(" where Id = ");
        cmdText.Append(this.id);
      }
      base.ExecuteNonQuery(base.GetCommand(cmdText.ToString()));
    }

    public void Delete(int id)
    {
      StringBuilder cmdText = new StringBuilder();
      cmdText.Append("Delete from Pensioners where Id = " + id);
      base.ExecuteNonQuery(base.GetCommand(cmdText.ToString()));
    }


  }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;

namespace PensionPaymentSystem
{
  public partial class PensionerManager : Form
```

```
 {
     public static string number;
     public static bool usePensionId = false;
     public PensionerManager()
     {
         InitializeComponent();
     }

     public void MapPensioner(Pensioner pensioner)
     {
         this.accountNumTextBox.Text = pensioner.AccountNumber;
         this.annualPayTextBox.Text = pensioner.AnnualPay;
         this.annualPayTextBox.Text = pensioner.AnnualPay;
         this.banksComboBox.Text = pensioner.Bank;
         try
         {
             this.birthDayDateTimePicker.Value = DateTime.Parse(pensioner.Birthday);
         }
         catch { }
         this.pensionerGroupBox.Text = pensioner.ControlNumber;
         try
         {
             this.dateOfEmpDateTimePicker.Value = DateTime.Parse(pensioner.DateOfEmployment);
         }
         catch { }
         try
         {
             this.dateOfRetDateTimePicker.Value = DateTime.Parse(pensioner.DateOfRetirement);
         }
         catch { }

         this.firstNameTextBox.Text = pensioner.FirstName;
         this.gradeLevelTextBox.Text = pensioner.GradeLevel;
         this.grossPayTextBox.Text = pensioner.GrossPay;
         this.lastDeptTextBox.Text = pensioner.LastDepartment;
         this.lastNameTextBox.Text = pensioner.LastName;
         this.monthlyPayTextBox.Text = pensioner.MonthlyPay;
         this.pensionIdTextBox.Text = pensioner.PensionId;
         this.rankTextBox.Text = pensioner.Rank;
         this.arrearsTextBox.Text = pensioner.Arrears;
         if (!string.IsNullOrEmpty(pensioner.Picture))
             this.pictureBox1.Image = Image.FromFile(Path.GetDirectoryName(Application.ExecutablePath)
+ "\\" + pensioner.Picture);
         else
             this.pictureBox1.Image = null;
     }

     private void PensionerManager_Load(object sender, EventArgs e)
     {
     }

     void tBox_KeyUp(object sender, KeyEventArgs e)
     {
         TextBox tBox = sender as TextBox;
         string text = tBox.Text;

         if (text != "")
         {
             try
             {
                 if (text.Length == 1)
                 {
                        if (text == "." || text == "+" || text == "-")
                            return;
                        try
                        {
                            float.Parse(text);
                        }
                        catch
                        {
                            throw;
                        }
                    }
                    else
                    {
                        if (text == "-." || text == "+.")
                            return;
                        try
                        {
                            float.Parse(text);
                        }
                        catch
                        {
                            throw;
                        }
                    }
                    this.grossPayTextBox.Text =
Pensioner.CalculateGrossPay(double.Parse(this.annualPayTextBox.Text)).ToString();
                    this.monthlyPayTextBox.Text =
Pensioner.CalculateMonthlyPay(double.Parse(this.grossPayTextBox.Text)).ToString();
                }
                catch { MessageBox.Show("Enter a numerial value"); tBox.Text = ""; }
            }
        }

        private void saveButton_Click(object sender, EventArgs e)
        {
            try
            {
                Pensioner pensioner = new Pensioner();
                if (!string.IsNullOrEmpty(this.pensionerGroupBox.Text))
                    pensioner.Retrieve(this.pensionerGroupBox.Text, true);

                pensioner.AccountNumber = this.accountNumTextBox.Text;
                pensioner.AnnualPay = this.annualPayTextBox.Text;
                pensioner.Arrears = this.arrearsTextBox.Text;
                pensioner.Bank = this.banksComboBox.Text;
                pensioner.FirstName = this.firstNameTextBox.Text;
                pensioner.LastName = this.lastNameTextBox.Text;
                try
                {
                    pensioner.Birthday = this.birthDayDateTimePicker.Value.ToShortDateString();
                }
                catch { }
                //
                // TODO: Generate Control Number
                //
                if (pensioner.Id == 0)
                {
                    string cNo = Guid.NewGuid().ToString();
                    try
                    {
                        pensioner.ControlNumber = pensioner.FirstName[0].ToString() +
pensioner.LastName[0].ToString() +
                            cNo.Substring(0, 8);
                    }
                    catch
                    {
                        MessageBox.Show("Lastname and firstname is required");
                        return;
                    }
                }
                //
                try
                {
                    pensioner.DateOfEmployment = this.dateOfEmpDateTimePicker.Value.ToShortDateString();
                }
                catch { }
                try
                {
                    pensioner.DateOfRetirement = this.dateOfRetDateTimePicker.Value.ToShortDateString();
                }
                catch { }
                pensioner.GradeLevel = this.gradeLevelTextBox.Text;
                pensioner.LastDepartment = this.lastDeptTextBox.Text;
                pensioner.PensionId = this.pensionIdTextBox.Text;
                pensioner.Rank = this.rankTextBox.Text;
                pensioner.Picture = string.IsNullOrEmpty(pensioner.Picture) ? "" : pensioner.Picture;
                if (!string.IsNullOrEmpty(this.fileNameTextBox.Text))
                {
                    //System.IO.MemoryStream ms = new System.IO.MemoryStream();
                    //Image pic = Image.FromFile(this.fileNameTextBox.Text);
                    //pic.Save(ms, System.Drawing.Imaging.ImageFormat.Jpeg);
                    if (File.Exists(this.fileNameTextBox.Text))
                    {
                        string filePath = Path.GetDirectoryName(Application.ExecutablePath) + "/Pictures/" +
Path.GetFileName(this.fileNameTextBox.Text);
                        File.Copy(this.fileNameTextBox.Text, filePath, true);
                        pensioner.Picture = "Pictures\\" + Path.GetFileName(filePath);
                    }
                }

                pensioner.Save();
                this.GetAll();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }

        }

        private void getAllButton_Click(object sender, EventArgs e)
        {
            this.GetAll();
        }

        void GetAll()
        {
            List<Pensioner> penList = Pensioner.Get("");
            this.pensionerBindingSource.Clear();
            foreach (Pensioner ps in penList)
                this.pensionerBindingSource.Add(ps);
        }

        private void dataGridView1_SelectionChanged(object sender, EventArgs e)
        {
            if (this.dataGridView1.CurrentRow != null)
                this.MapPensioner(this.dataGridView1.CurrentRow.DataBoundItem as Pensioner);
        }

        private void pictureButton_Click(object sender, EventArgs e)
        {
            this.openFileDialog1.Filter = "Jpg Files|*.jpg|Jpeg Files|*.jpeg|Gif files|*.gif|All Files|*.*";
            this.openFileDialog1.ShowDialog();
        }

        private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
        {
            this.fileNameTextBox.Text = this.openFileDialog1.FileName;
        }

        private void addButton_Click(object sender, EventArgs e)
        {
            this.MapPensioner(new Pensioner());
        }

        private void deleteButton_Click(object sender, EventArgs e)
        {
            if (this.dataGridView1.CurrentRow != null)
                new Pensioner().Delete(((Pensioner)this.dataGridView1.CurrentRow.DataBoundItem).Id);
            this.GetAll();
        }

        private void Get(string whereClause)
        {
            List<Pensioner> penList = Pensioner.Get(whereClause);
            this.pensionerBindingSource.Clear();
            foreach (Pensioner ps in penList)
                this.pensionerBindingSource.Add(ps);
        }

        private void nextButton_Click(object sender, EventArgs e)
        {
            if (this.dataGridView1.CurrentRow != null)
            {
                int lastIndex = this.dataGridView1.Rows.Count - 1;
                int currentIndex = this.dataGridView1.CurrentRow.Index;
                if (currentIndex < lastIndex)
                {
                    this.dataGridView1.CurrentCell =
this.dataGridView1[this.dataGridView1.CurrentCell.ColumnIndex,
                        this.dataGridView1.CurrentCell.RowIndex + 1];
                    this.dataGridView1_SelectionChanged(null, null);
                }
            }
        }

        private void previousButton_Click(object sender, EventArgs e)
        {
            if (this.dataGridView1.CurrentRow != null)
            {
                //int lastIndex = this.dataGridView1.Rows.Count - 1;
                int currentIndex = this.dataGridView1.CurrentRow.Index;
                if (currentIndex > 0)
                {
                    this.dataGridView1.CurrentCell =
this.dataGridView1[this.dataGridView1.CurrentCell.ColumnIndex,
                        this.dataGridView1.CurrentCell.RowIndex - 1];
                    this.dataGridView1_SelectionChanged(null, null);
                }
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            if(this.dataGridView1.CurrentRow != null)
                this.MapPensioner(this.dataGridView1.CurrentRow.DataBoundItem as Pensioner);
        }

        private void button4_Click(object sender, EventArgs e)
        {
            Search search = new Search();
            search.FormClosing += new FormClosingEventHandler(search_FormClosing);
            search.ShowDialog();
        }

        void search_FormClosing(object sender, FormClosingEventArgs e)
        {
            Search search = sender as Search;
            if (search.search)
            {
                string column = PensionerManager.usePensionId ? "PensionId" : "ControlNumber";
                this.Get(column + " = '" + PensionerManager.number + "'");
            }
        }

        private void printButton_Click(object sender, EventArgs e)
        {
            Report rp = new Report(this.pensionerBindingSource);
            rp.MdiParent = this.MdiParent;
            rp.Show();
        }


    }
}
```